



(19)

(11) Publication number: **10301846 A**

Generated Document.

## PATENT ABSTRACTS OF JAPAN

(21) Application number: **10097589**

(51) Intl. Cl.: **G06F 12/08 G06F 12/16**

(22) Application date: **09.04.98**

(30) Priority:	<b>14.04.97 US 97 839554</b>	(71) Applicant: <b>INTERNATL BUSINESS MACH CORP &lt;IBM&gt;</b>
(43) Date of application publication:	<b>13.11.98</b>	(72) Inventor: <b>RAVI KUMAR ARIMIRI JOHN STEPHEN DODDSON JERRY DON LEWIS TIMOTHY M SUKAAGAN</b>
(84) Designated contracting states:		(74) Representative:

### (54) **FUNCTIONAL BYPASS METHOD AND SYSTEM FOR CACHE ARRAY DEFECT USING RESTORATION MASK**

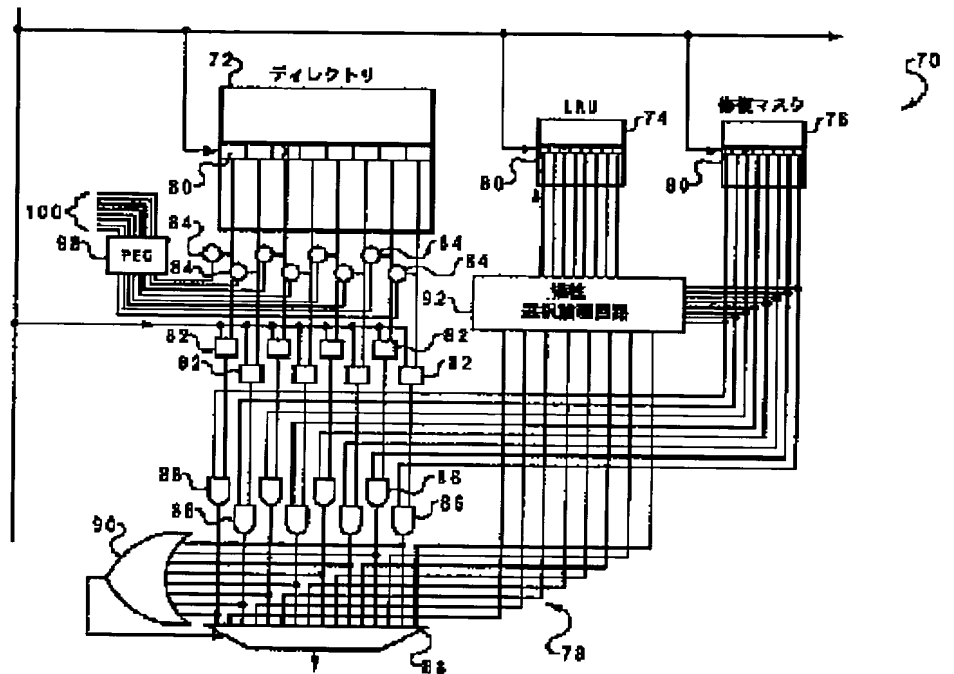
(57) Abstract:

PROBLEM TO BE SOLVED: To bypass a defect inside a cache used by the processor of a computer system by using a restoration mask, preventing a defective cache line from becoming a cache hit and preventing the defective cache line from being selected as a sacrifice for cache replacement.

SOLUTION: This system is provided with the restoration mask 76 provided with the array of bit fields each one of which corresponds to each one of plural cache lines inside the cache. A specified cache line inside the cache is identified as the defective one. The corresponding bit field inside the array of the restoration mask 76 is set

and it is indicated that the defect is present in the defective cache line. Based on the corresponding bit field inside the array of the restoration mask 76, access to the defective cache line is prevented. By executing the steps, the defect inside the cache is bypassed.

COPYRIGHT: (C)1998,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-301846

(43) 公開日 平成10年(1998)11月13日

(51) Int.Cl.<sup>6</sup>  
G 0 6 F 12/08

識別記号

12/16

3 1 0

F I  
G 0 6 F 12/08

12/16

J

E

3 1 0 E

審査請求 未請求 請求項の数16 O L (全 14 頁)

(21) 出願番号 特願平10-97589

(22) 出願日 平成10年(1998) 4 月 9 日

(31) 優先権主張番号 0 8 / 8 3 9 5 5 4

(32) 優先日 1997年 4 月 14 日

(33) 優先権主張国 米国 (U S)

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSIN  
ESS MACHINES CORPO  
RATION

アメリカ合衆国10504、ニューヨーク州  
アーモンク (番地なし)

(72) 発明者 ラビ・クマール・アリミリ

アメリカ合衆国78759 テキサス州オース  
チン スパイスブラッシュ・ドライブ  
9221

(74) 代理人 弁理士 坂口 博 (外 1 名)

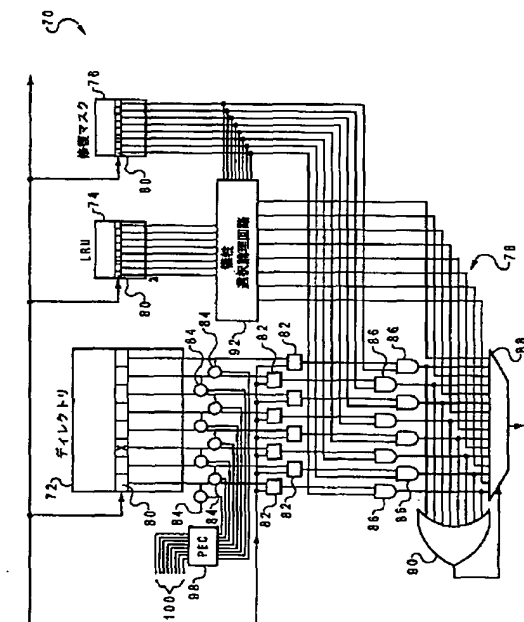
最終頁に続く

(54) 【発明の名称】 修復マスクを使用したキャッシュ・アレイ欠陥の機能的迂回方法及びシステム

(57) 【要約】

【課題】 コンピュータ・システムのプロセッサによって使用されるキャッシュ内の欠陥を迂回する方法。

【解決手段】 修復マスクが、キャッシュ内のキャッシュ・ラインに対応するビット・フィールドの配列を有し、キャッシュ内の特定のキャッシュ・ラインが欠陥があるものと識別されると、その特定のキャッシュ・ラインに欠陥があることを示すように修復マスク配列内の対応するビット・フィールドが設定され、修復マスク配列内の対応するビット・フィールドに基づいて、欠陥のあるキャッシュ・ラインへのそれ以上のアクセスが防止される。この修復マスクを使用して、欠陥のあるキャッシュ・ラインが決してキャッシュ・ヒットにならないように防止することができ、欠陥のあるキャッシュ・ラインがキャッシュ置換のための犠牲として決して選択されないように防止することができる。セット・アソシアティブ・キャッシュを使用し、それによって欠陥のあるキャッシュ・ラインをそれぞれの合同クラスから実質的に除去する。



【特許請求の範囲】

【請求項1】 コンピュータ・システムのプロセッサによって使用されるキャッシュにおいて欠陥を迂回する方法であって、

各ビット・フィールドがキャッシュ内の複数のキャッシュ・ラインのうちのそれぞれ1つのキャッシュ・ラインに対応するビット・フィールドの配列を有する修復マスクを設けるステップと、

キャッシュ内の特定のキャッシュ・ラインに欠陥があるものとして識別するステップと、

修復マスク配列内の対応するビット・フィールドを設定して特定のキャッシュ・ラインに欠陥があることを示すステップと、

修復マスク配列内の対応するビット・フィールドに基づいて欠陥のあるキャッシュ・ラインへのアクセスを防止するステップとを含む方法。

【請求項2】 防止する前記ステップが、欠陥のあるキャッシュ・ラインが決してキャッシュ・ヒットにならないように防止するステップを含む、請求項1に記載の方法。

【請求項3】 欠陥のあるキャッシュ・ラインが決してキャッシュ置換のための犠牲として選択されないように防止するステップを含む、請求項1に記載の方法。

【請求項4】 各合同クラスが複数のキャッシュ・ラインからなる複数の合同クラスのうちの1つに所与のメモリ・ブロックをマップするステップをさらに含み、欠陥のあるキャッシュ・ラインへのアクセスを防止する前記ステップの結果、欠陥のあるキャッシュ・ラインがそれぞれの合同クラスから実質的に除去される、請求項1に記載の方法。

【請求項5】 欠陥のあるキャッシュ・ラインへのアクセスを防止する前記ステップによって、キャッシュが、冗長性をもたせるためにキャッシュ・ラインを確保しておくことなく欠陥のないすべてのキャッシュ・ラインを使用することができるようにする、請求項1に記載の方法。

【請求項6】 キャッシュが、要求されたアドレスがキャッシュ・ラインの一部にあるアドレス・タグに対応するかどうかを判断する複数の比較器を有し、所与の比較器が、要求されたアドレスが前記アドレス・タグ部分と一致しない場合はイナクティブであるが、要求されたアドレスが前記アドレス・タグ部分と一致する場合はアクティブであり、

欠陥のあるキャッシュ・ラインが決してキャッシュ・ヒットにならないように防止する前記ステップが、各信号が比較器のうちの所与の1つの比較器の出力と修復マスクからのビット・フィールドのうちのそれぞれ1つのビット・フィールドの出力とのAND結合である複数の信号を入力信号として受け取るマルチプレクサを設けるステップを含む、請求項2に記載の方法。

【請求項7】 キャッシュが、複数のキャッシュ・ラインのうちのどのキャッシュ・ラインを置換のための暫定的犠牲として選択するかを示す複数の出力信号を有するキャッシュ置換回路を有し、

欠陥のあるキャッシュ・ラインが犠牲として選択されないように防止する前記ステップが、複数の信号を入力信号として受け取るマルチプレクサを設けるステップを含み、各信号が、暫定犠牲に欠陥がないように保証するために修復マスクのビット・フィールドから入力信号を受け取る代替犠牲選択論理回路によって変更されたキャッシュ置換回路の出力信号である、請求項3に記載の方法。

【請求項8】 防止する前記ステップが、欠陥のあるキャッシュ・ラインが決してキャッシュ・ヒットにならないように防止するステップをさらに含む、請求項3に記載の方法。

【請求項9】 プロセッサと、メモリ・デバイスと、前記プロセッサと前記メモリ・デバイスとに接続され、前記メモリ・デバイスのアドレスに対応するメモリ・ブロックを記憶する複数のキャッシュラインを有するキャッシュと、

各ビット・フィールドが、選択されたキャッシュ・ラインに欠陥があることを示す値を有する前記選択されたキャッシュ・ラインに対応する前記ビット・フィールドのうちの所与の1つのビット・フィールドに基づいて前記複数のキャッシュ・ラインのうちの選択された1つのキャッシュ・ラインへのアクセスを防止する、前記キャッシュ・ラインのうちのそれぞれ1つのキャッシュ・ラインに対応する複数のビット・フィールドを有する修復マスクとを含むコンピュータ・システム。

【請求項10】 前記修復マスク手段が、欠陥のあるキャッシュ・ラインが決してキャッシュ・ヒットにならないように防止する、請求項9に記載のコンピュータ・システム。

【請求項11】 前記修復マスク手段が、欠陥のあるキャッシュ・ラインが決してキャッシュ置換のための犠牲として選択されないように防止する、請求項9に記載のコンピュータ・システム。

【請求項12】 前記キャッシュが、各合同クラスが複数の前記キャッシュ・ラインから成る複数の合同クラスの1つに所与のメモリ・ブロックをマップし、前記修復マスク手段が、欠陥のあるキャッシュ・ラインをその対応する合同クラスから有効に除去することにより、その欠陥のあるキャッシュ・ラインへのアクセスを防止する、請求項9に記載のコンピュータ・システム。

【請求項13】 前記修復マスク手段によって、前記キャッシュがキャッシュ・ラインを冗長性をもたせるために確保しておくことなくすべての欠陥のないキャッシュ・ラインを使用することができるようにする、請求項9に

記載のコンピュータ・システム。

【請求項 14】前記キャッシュが、要求されたアドレスがキャッシュ・ラインの一部にあるアドレス・タグに対応するかどうかを判断する複数の比較器を有し、前記比較器のうちの所与の 1 つの比較器が、前記要求されたアドレスが前記アドレス・タグ部分と一致しない場合はイナクティブであるが、前記要求されたアドレスが前記アドレス・タグ部分と一致する場合はアクティブである出力信号を有し、前記キャッシュが、複数の信号を受け取り、各信号が前記比較器の前記出力信号の 1 つと前記修復マスク手段から前記ビット・フィールドのうちのそれぞれ 1 つのビット・フィールドの出力信号との AND 結合である複数の信号を入力信号として受け取るマルチプレクサをさらに含む、請求項 10 に記載のコンピュータ・システム。

【請求項 15】前記キャッシュが、複数の前記キャッシュ・ラインのうちのどのキャッシュ・ラインを置換のための暫定的犠牲として選択するかを示す複数の出力信号を有するキャッシュ置換回路を有し、各信号が前記修復マスク手段の前記ビット・フィールドから入力信号を受け取る代替犠牲選択論理回路によって変更された前記キャッシュ置換回路の出力信号を有する複数の信号を入力信号として受け取るマルチプレクサをさらに含む、請求項 11 に記載のコンピュータ・システム。

【請求項 16】前記修復マスク手段が欠陥のあるキャッシュ・ラインが決してキャッシュ・ヒットにならないようにさらに防止する、請求項 11 に記載のコンピュータ・システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は一般にはコンピュータ・システムに関し、具体的にはコンピュータ・システムのプロセッサが使用するキャッシュのパフォーマンスを向上させる方法に係わる。

【0002】

【従来の技術】従来のコンピュータ・システム 10 の基本構造を図 1 に示す。コンピュータ・システム 10 は、1 つまたは複数の処理ユニットを有することができ、そのうちの 2 つ 12 a 及び 12 b が図示されている。処理装置は、入出力 (I/O) 装置 14 (表示モニタ、キーボード、永続記憶装置など)、処理装置がプログラム命令を実行するために使用するメモリ・デバイス 16 (ランダム・アクセス・メモリ、または RAM)、及び、コンピュータに最初に電源を入れたときに周辺装置の 1 つ (通常は永続記憶装置) からオペレーティング・システムを探し出してロードすることを主な目的とするファームウェア 18 を含む様々な周辺装置に接続されている。処理装置 12 a 及び 12 b は、汎用相互接続機構またはバス 20 を含む様々な手段によって周辺装置と通信する。コンピュータ・システム 10 は、たとえばモデムや

プリンタなどに接続するためのシリアル・ポートやパラレル・ポートなど、図示されていない多くの追加の構成要素を有することができる。当業者ならさらに、図 1 のブロック図に図示されている構成要素と共に使用可能な他の構成要素もあることがわかるであろう。たとえば、ビデオ表示モニタを制御するためにディスプレイ・アダプタを使用したり、メモリ 16 にアクセスするためにメモリ・コントローラを使用することができる。また、I/O 装置 14 をバス 20 に直接接続する代わりに、二次 (I/O) バスに接続することもでき、二次バスはさらに I/O ブリッジに接続され、I/O ブリッジはバス 20 に接続される。コンピュータは 3 個以上の処理装置を有することもできる。

【0003】対称マルチプロセッサ (SMP) コンピュータでは、すべての処理装置は一般に同じである。すなわち、すべての処理装置が命令及びプロトコルの共通のセットまたはサブセットを使用して動作し、一般に同じアーキテクチャを有する。典型的なアーキテクチャは図 1 に示す通りである。処理装置は、複数のレジスタと、コンピュータを動作させるためにプログラム命令を実行する実行ユニットとを含むプロセッサ・コア 22 を含む。処理装置の例としては、インターナショナル・ビジネス・マシーンズ・コーポレーション (IBM) が販売する Power PC™ プロセッサがある。処理装置は、高速メモリ・デバイスを使用して実装された命令キャッシュ 24 やデータ・キャッシュ 26 などの 1 つまたは複数のキャッシュも有することができる。キャッシュは、メモリ 16 から値をロードする長いステップを回避することによって処理を高速化する目的で、プロセッサが繰り返しアクセスする可能性のある値を一時的に記憶するために一般に使用される。これらのキャッシュは、単一の集積チップ 28 上にプロセッサ・コアと一体にパッケージされているときは、「オンボード」キャッシュと呼ばれる。各キャッシュには、プロセッサ・コアとキャッシュ・メモリとの間のデータの転送を管理するキャッシュ・コントローラ (図示せず) が付随している。

【0004】処理装置 12 は、オンボード (一次) キャッシュ 24 及び 26 をサポートするため二次キャッシュ (L2) と呼ばれるキャッシュ 30 などの追加のキャッシュを備えることができる。言い換えると、キャッシュ 30 はメモリ 16 とオンボード・キャッシュとの間の媒介として機能し、オンボード・キャッシュよりもはるかに大量の情報 (命令及びデータ) を記憶することができるが、アクセス・ペナルティは長い。たとえば、キャッシュ 30 は、256 キロバイトまたは 512 キロバイトの記憶容量を有するチップとすることができ、プロセッサは 64 キロバイトの合計記憶容量を持つオンボード・キャッシュを有する IBM Power PC™ 604 シリーズのプロセッサとすることができる。キャッシュ 30 はバス 20 に接続され、メモリ 16 からプロセッサ・

コア 2 2 への情報のロードはすべてキャッシュ 3 0 を介して行わなければならない。図 1 には二次キャッシュ階層しか図示されていないが、多くのレベルの相互接続されたキャッシュを備えたマルチレベル・キャッシュ階層を設けることもできる。

【0005】キャッシュは、様々な命令及びデータ値を個別に記憶する多くの「ブロック」を有する。どのキャッシュ内のブロックも「セット」または「合同クラス」と呼ばれるブロックのグループに分けられている。セットとは、その中に所与のメモリ・ブロックが入ることができるキャッシュ・ブロックの集まりである。どの所与のメモリ・ブロックについても、事前設定マッピング機能に従ってブロックをマップすることができるキャッシュ内の固有のセットがある。セット内のブロック数をキャッシュのアソシアティビティと呼び、たとえば 2 ウェイ・セット・アソシアティブとは、所与のメモリ・ブロックについてキャッシュ内にそのメモリ・ブロックをマップすることができるブロックが 2 個あることを意味する。しかし、所与のセットにメイン・メモリ内のいくつかの異なるブロックをマップすることができる。1 ウェイ・セット・アソシアティブ・キャッシュにはダイレクト・マップされる。すなわち、特定のメモリ・ブロックを含むことができるキャッシュ・ブロックは 1 つしかない。メモリ・ブロックがいずれかのキャッシュ・ブロックを専有することができる場合、すなわち 1 つの合同クラスがあり、アドレス・タグがメモリ・ブロックのフル・アドレスである場合、キャッシュはフル・アソシアティブであると言う。

【0006】キャッシュ・ライン（ブロック）の例としては、アドレス・タグ・フィールド、状態ビット・フィールド、包含ビット・フィールド、実際の命令及びデータを記憶する値フィールドがある。状態ビット・フィールドと包含ビット・フィールドは、マイクロプロセッサ・コンピュータ・システム内のキャッシュ・コヒーレンスを維持するために使用される（キャッシュに記憶されている値の有効性を示す）。アドレス・タグは、対応するメモリ・ブロックのフル・アドレスのサブセットである。着信アドレスとアドレス・タグ・フィールド内のタグの 1 つとの比較一致によって、キャッシュ「ヒット」が示される。キャッシュ内のすべてのアドレス・タグ（及び場合によっては状態ビット・フィールドと包含ビット・フィールド）の集まりをディレクトリと呼び、すべての値フィールドの集まりはキャッシュ項目配列である。

【0007】所与のクラスの合同クラス内のすべてのブロックがいっぱいであり、そのキャッシュがいっぱいになった合同クラスにマップする記憶場所に対する「読取り」または「書込み」要求を受け取った場合、キャッシュは現在そのクラスに入っているブロックの 1 つを「追い出し」しなければならない。キャッシュは、当業者に

周知のいくつかの手段（最低使用頻度（LRU）法、ランダム法、疑似 LRU 法など）の 1 つによって、追い出すブロックを選択する。選択されたブロック内のデータが変更された場合、そのデータはメモリ階層内の次に低いレベルに書き込まれる。これは他のキャッシュ（この場合は一次キャッシュすなわちオンボード・キャッシュ）またはメイン・メモリ（この場合は図 1 の 2 レベル・アーキテクチャに図示するような二次キャッシュ）でもよい。包含の原理により、階層の下位レベルは書き込まれた変更データを保持するために使用可能なブロックをすでに持っていることになる。しかし、選択されたブロック内のデータが変更されなかった場合、そのブロックは単に放棄されるだけで、階層内の次に低いレベルには書き込まれない。階層の 1 つのレベルからブロックを除去するこのプロセスを「追い出し」と呼ぶ。このプロセスの終わりに、キャッシュは追い出されたブロックのコピーをもちや保持しない。

【0008】図 2 に、上述のキャッシュ構造と追い出しプロセスを示す。キャッシュ 4 0（一次またはそれより下位レベル）は、キャッシュ・ディレクトリ 4 2 と、キャッシュ項目配列と 4 4 と、LRU 配列 4 6 と、特定の合同クラスから追い出すブロックを選択する制御論理回路 4 8 とを含む。図のキャッシュ 4 0 は 8 ウェイ・セット・アソシアティブであり、したがってディレクトリ 4 2 とキャッシュ項目配列 4 4 と LRU 配列 4 6 とはそれぞれ、5 0 に示すように特定の合同クラスについて 8 ブロックから成る特定のセットを有する。言い換えると、合同クラス 5 0 の「X」で示すようにキャッシュ・ディレクトリ 4 2 内の合同クラスの特定のメンバには、キャッシュ項目配列 4 4 内の合同クラスの特定のメンバと、LRU 配列 4 6 内の合同クラスの特定のメンバとが関連づけられている。

【0009】ディレクトリ 4 2 内の各ブロックは誤り修正コード（ECC）回路 5 2 を介して制御論理回路に接続されている。ソフト誤り（漂遊放射や静電放電など）またはハード誤り（欠陥セル）のために、所与のキャッシュ・ブロック内のビットに不正な値が入っていることがある。ECC を使用すれば適正なデータ・ストリームを再構築することができる。ある種の ECC は単一ビット誤りの検出と修正にのみ使用することができる。すなわち、特定のブロック内の 2 つ以上のビットが無効な場合、その ECC は適正なデータ・ストリームは実際にはどのようなストリームであるかを判断することはできないが、少なくともその欠陥は検出することができる。また、二重ビット誤りの検出や修正も行うことができるより高度な ECC もある。この二重ビット誤りは修正するのにコストがかかるが、その設計は二重ビット誤りが発生すると機械を停止させるものである。ディレクトリ 4 2 のみ ECC 回路を有するように図示されているが、これらの回路はキャッシュ項目配列 4 4 など、他の配列で

も同様に使用することができる。

【0010】（修正された）メモリ・ブロック・アドレスに対応する値を持つECC回路52の出力がそれぞれの比較器54に接続され、各比較器は要求されたメモリ・ブロックのアドレスも受け取る。要求されたメモリ・ブロックの有効なコピーが合同クラス50内にある場合、比較器54のうちの1つの比較器だけがアクティブ信号を出力する。比較器54の出力はマルチプレクサ56に接続され、ORゲート58にも接続されている。ORゲート58の出力はマルチプレクサ56を制御する。キャッシュ・ヒットが起こると（要求されたアドレスがキャッシュ・ディレクトリ42内のアドレスと一致する）、ORゲート58はマルチプレクサ56をアクティブにして合同クラスのどのメンバがそのアドレスと一致するかを示す信号を渡す。この信号は、キャッシュ項目配列44内の各項目から入力値を受け取るもう一つのマルチプレクサ60を制御する。このようにして、ディレクトリ内でキャッシュ・ヒットが起こると、それに対応する値がマルチプレクサ60からバス62に流される。

【0011】キャッシュ・ミスが起こったとき、特定の合同クラス50内のすべてのブロックがメモリ・ブロックの有効なコピーをすでに持っている場合は、合同クラス50内のキャッシュ・ブロックのうちの1つを選択して犠牲にしなければならない。この選択はLRU配列46内の合同クラスのLRUビットを使用して行われる。クラス内の各キャッシュ・ブロックについて複数のLRUビットがある。たとえば、8ウェイ・アソシアティブ・キャッシュの場合は1ブロックにつき3つのLRUビットがある。クラス内の各ブロックからLRUビットがデコード64に入力値として供給される。デコードはどのブロックを犠牲にするかを示す8ビットの出力値を有する。この出力値はマルチプレクサ56に結合される。このようにして、ORゲート58がアクティブでない場合、マルチプレクサ56はデコード64の出力に基づいて使用するキャッシュ・ブロックを示す標識を渡す。

【0012】上述のECC回路はメモリ・セル内で発生したソフト誤りを処理する1つの方法である。ハード誤りを処理する他の手法は、配列（ディレクトリ、LRU、キャッシュ）内に冗長性をもたせることである。キャッシュ・チップを製作するとき、各配列内に欠陥のある行ラインまたは列ラインがないかどうか検査して調べることができる（キャッシュ、ディレクトリ、及びLRU全体について行ライン及び列ラインを検査する）。配列に欠陥がある場合、ヒューズを永続的に切ってその欠陥性を示すことができる。次に、各アクセスされるアドレスについて配列内部で比較を行い、欠陥のあるアドレスと一致していないか調べる。一致している場合、適切な論理回路がそのアドレスを、チップ上に形成された多くの余分の行ライン及び列ライン（すなわち冗長ビット・ライン（列）及びワード・ライン（行）で形成された

ライン）の1つに経路指定し直す。余分のビット・ライン及びワード・ラインの数は欠陥率と所望のチップ歩留まりとによって異なることがある。低欠陥（物理サイズが比較的大きい）キャッシュの場合、正規のライン256本ごとに2本ずつの余分なラインを設け、高欠陥（物理サイズが比較的小さい）キャッシュでは正規のライン8本ごとに2本の余分のラインを設ける。

【0013】上述のキャッシュ構成にはいくつかの欠陥と制約がある。ECC回路52については、その種の回路はかなり複雑で、チップ上のスペースをとるだけでなく、キャッシュ値を（ディレクトリまたはキャッシュから）取り出すためのクリティカル（タイミング）パス内にあるため処理速度がさらに遅くなる。ECC回路は二重ビット誤りの修正は可能であるかも知れないが、2ビットを超える不正ビットのある複数ビット誤りは修正することができない。複雑さを増し、処理速度を低下させるこれらの従来技術のキャッシュ構成のもう一つの面は、キャッシュをCPUスヌープのためにCPUに相互接続したりシステム・バス・スヌープのためにシステム・バスに相互接続したりを選択的に行うために必要なアービトレーション論理回路66である。この論理回路もクリティカル・パスにある。このように2つのスヌーピング装置が1つのキャッシュを使用すると、本質的に、CPUとシステム・バスの両方が同時にそのキャッシュの読取りを行う必要がある場合など、他の特定の遅延を生じさせる。この2つの読取り操作を同時に行うことはできず、アービトレーション論理回路によって順次化しなければならない。これと同じことは書込み操作にも言える。

【0014】キャッシュ構成の他の欠点は、ビット・ライン冗長構成とワード・ライン冗長構成の使用に関係する。この技法によってチップ歩留まりを向上させることはできるが、これらの冗長構成は配列アクセスのためのクリティカル・パス内に直接入る。欠陥のある行ライン及び列ラインを探索し、欠陥のあるラインと一致する要求をリダイレクトするのに余分な時間を要する。したがって、歩留まり向上とキャッシュ応答速度の低下との間にトレードオフがある。他の欠点は冗長ラインのために用意しなければならない追加の物理サイズである。正規ライン8本ごとに2本の余分なラインを備える例では、25%の余分なキャッシュ・サイズ（オーバーヘッド）が必要であり、その空間の多くは一度も使用されない。また、冗長ラインはあまりうまく拡張されない。たとえば、キャッシュ・ライン・サイズが2倍（64バイトから128バイトに）になった場合、冗長ラインに必要なシリコンの量（チップ空間）も同様に2倍になる。最後に、キャッシュ内のヒューズ切れに基づいて使用される冗長ラインの使用は静的であり、かなり無駄がある。冗長ラインの可用性は検査時のキャッシュの状態に基づく。接合部の温度と内部電圧が変化する条件の

下で動作する高密度の大規模なキャッシュ・チップでは、キャッシュ内の欠陥が増大するが、それらの追加の欠陥ラインをリダイレクトすることができない。歩留まりを向上させるために多くの余分なラインを設けなければならないが、それらのラインの多くは一度も使用されない。

【0015】上記に鑑みて、キャッシュ・アクセスの高速化と例外誤り修正機能とを含む、欠陥キャッシュ・ラインの改良型処理機能を有するキャッシュ構成を備えることが望ましいであろう。さらに、複雑な論理回路を設けずにすべての使用可能なキャッシュ・ラインを効率的かつ動的に使用し、キャッシュ・ラインの拡張に合わせて適切に拡張されるようなキャッシュ構成を備えれば有利であろう。

【0016】

【発明が解決しようとする課題】したがって、本発明の目的は、コンピュータ・システムのプロセッサが使用する改良されたキャッシュを提供することである。

【0017】本発明の他の目的は、クリティカル・パスに余分の論理回路のない、すべての使用可能なキャッシュ・ラインを効率的に使用する前記キャッシュを提供することである。

【0018】本発明の他の目的は、欠陥回避及び誤り修正を含む、欠陥の改良型処理機能を有する前記キャッシュを提供することである。

【0019】本発明の他の目的は、より高速の読取りアクセス機能を有する前記キャッシュを提供することである。

【0020】

【課題を解決するための手段】上記の目的は、コンピュータ・システムのプロセッサが使用するキャッシュ内の欠陥を迂回する方法であって、一般に、各ビット・フィールドがキャッシュ内の複数のキャッシュ・ラインのうちのそれぞれ1つのキャッシュ・ラインに対応するビット・フィールドの配列を有する修復マスクを設けるステップと、キャッシュ内の特定のキャッシュ・ラインを欠陥があるものと識別するステップと、修復マスク配列内の対応するビット・フィールドを設定して欠陥キャッシュ・ラインに欠陥があることを示すステップと、修復マスク配列内の対応するビット・フィールドに基づいて欠陥キャッシュ・ラインへのアクセスを防ぐステップとを含む方法で達成される。修復マスクを使用して、欠陥キャッシュ・ラインがキャッシュ・ヒットになるのを防ぎ、欠陥キャッシュ・ラインがキャッシュ置換のための犠牲として選択されるのを防ぐことができる。セット・アソシアティブ・キャッシュを使用して、欠陥キャッシュ・ラインをそれぞれの合同クラスから実質的に除去する。この手法によって、キャッシュは欠陥のないすべてのキャッシュ・ラインを使用することができ、冗長性をもたせるために確保しておかれるキャッシュ・ラインは

ない。一実施例では、キャッシュは、要求されたアドレスがキャッシュ・ラインの一部にあるアドレス・タグに対応するかどうかを判断する複数の比較器を有し、所与の比較器は、要求されたアドレスがアドレス・タブ部分と一致しない場合はイナクティブであるが、要求されたアドレスがアドレス・タブ部分と一致する場合はアクティブである出力信号を有し、欠陥キャッシュ・ラインがキャッシュ・ヒットになるのを防止する前記ステップが、各信号が比較器のうちの所与の1つの比較器の出力信号と修復マスクからのビット・フィールドのそれぞれ1つのビット・フィールドの出力信号とのAND結合である複数の信号を入力信号として受け取るマルチプレクサを設けるステップを含む。キャッシュは、複数のキャッシュ・ラインのうちのいずれのキャッシュ・ラインを置換のための犠牲として選択するかを示す複数の出力信号を有するキャッシュ置換（追い出し）回路も有し、マルチプレクサが、各信号がキャッシュ置換回路の出力信号のうちの所与の1つの出力信号と修復マスクからのビット・フィールドのうちのそれぞれ1つのビット・フィールドの出力信号とのAND結合である他の複数の信号を入力信号として受け取る。

【0021】

【発明の実施の形態】図面、特に図3を参照すると、本発明により構成されたセット・アソシアティブ・キャッシュ70の一実施例を示す高水準略図が図示されている。キャッシュ70は一般に、キャッシュ・ディレクトリ72と、LRU配列74と、修復マスク76と、制御論理回路78とを含む。キャッシュ・ディレクトリ72は、関連づけられていくつかのセットになっている複数のアドレス・タグを有し、図の実施例は8ウェイ・セット・アソシアティブであるため、ディレクトリ72、LRU配列74、及び修復マスク76は図80に示すようにそれぞれ特定の合同クラスについて8ブロックからなる特定のセットを有する。キャッシュ項目配列（図示せず）はディレクトリ72内のアドレス・タグに関連づけられた値を有する。

【0022】ディレクトリ72の所与の合同クラス内の各ブロックは、それぞれの比較器82に接続された出力線を有し、各比較器は要求されたメモリ・ブロックのアドレスを受け取る。要求されたメモリ・ブロックの有効なコピーが合同クラス80内にある場合、比較器82のうちのただ1つの比較器がセットのどのメンバが有効なコピーを保持しているかを示すアクティブ信号を出力する。ディレクトリ72からの各出力線は、それぞれの比較器82との接続と並列してそれぞれのパリティ検査器84にも接続されている。パリティ検査器84は誤りビットの修正は行わず、ディレクトリ72からのアドレス・タグ（または状態ビット）の読取りの際に誤りが発生したことを示すためにのみ使用される。この誤り情報は以下で詳述するように使用される。重要なのは、パリティ



ィ検査器は比較器と並列して接続されているため、クリティカル・パスの外にあることである。すなわち、パリティ検査器は、残りの論理回路が要求を処理しているときに同時に並行してパリティ検査を行うことができるためキャッシュ・パフォーマンスに影響を与えない（すなわち速度を低下させない）。（当然ながら、誤りが発生したことが検出された場合はパリティ検査器によって速度が低下するが、これは例外である。）また、パリティ検査器は、従来のパリティ検査技法を使用することができ、従来技術で使用されているような誤り修正コード

（ECC）回路よりも複雑でなく（図2と比較）、したがってキャッシュ・チップ（シリコン・ウエハ）上の占有空間が少ない。キャッシュ・ディレクトリのためのオフライン・パリティ検査器の使用が図示されているが、さらにキャッシュ項目配列も従来のECC論理回路を設けずにこれを使用することができる。

【0023】比較器82の出力線はANDゲート86の配列に接続されている。各ANDゲート86は、1つの比較器出力信号と修復マスク76からもう一つの出力信号を受け取る。修復マスク76を使用して特定のブロック（ディレクトリ72、キャッシュ項目配列、またはLRU配列74内の部分を含む、キャッシュ・ラインの一部）に欠陥があるかどうかを示す。この実施例では、対応するキャッシュ・ラインが有効な場合、修復マスク76の出力の状態はアクティブ（または高）である。キャッシュ・ラインに欠陥がある場合、修復マスク78の対応する出力がイナクティブ（オフまたは低）になる。このようにして、ANDゲート86の出力は、（1）対応する比較器82がキャッシュ・ヒットを示しており、

（2）修復マスク76内の対応する項目によってラインが有効であることが示された場合にのみアクティブになる。言い換えると、修復マスク76内の対応する項目によってそのラインに欠陥があることが示されることを除けばキャッシュ・ヒットが起こることになる場合、修復マスク76内のその項目の出力はイナクティブになり、対応するANDゲート86の出力もイナクティブになる（「ミス比較」）。したがって、欠陥のあるキャッシュ・ラインがキャッシュ・ヒットになることは決してない。

【0024】ANDゲート86の出力信号はマルチプレクサ88に供給され、ORゲート90にも供給される。ORゲート90の出力信号によってマルチプレクサ88が制御される。欠陥のないキャッシュ・ラインでキャッシュ・ヒットが起こった場合、ORゲート90はマルチプレクサ88をアクティブにして、合同クラスのどのメンバがそのアドレスと一致するかを示す信号をキャッシュ項目配列に渡す。キャッシュ・ミスが起こった場合、及び特定の合同クラス80内のすべてのブロックがメモリ・ブロックの有効なコピーをすでに持っている場合、キャッシュ・ブロックの1つを選択して犠牲にしなけれ

ばならない。この選択は、従来のLRU（最長期間未使用）アルゴリズムを含むことができるLRU配列74によって暫定的に行われる。LRUの出力信号は、どのブロックが犠牲のために暫定的に選択されたかを示し、代替犠牲選択論理回路92に接続される。この選択回路92は修復マスク76からも入力信号を受け取る。暫定的に選択された犠牲ブロックに欠陥がない場合、代替犠牲選択論理回路92は単にLRU出力信号をマルチプレクサ88に渡し、マルチプレクサ88はその犠牲情報をキャッシュ項目配列に渡す。暫定的に選択された犠牲ブロックに欠陥がある場合、代替犠牲選択論理回路92は新しい暫定的犠牲を選択する。新しい暫定的犠牲に欠陥がない場合、それに対応する情報がマルチプレクサ88に送られる。新しい暫定的犠牲にも欠陥がある場合、代替犠牲選択論理回路92は、欠陥のない犠牲が選択されるまでこのプロセスを繰り返す。したがって、欠陥のあるキャッシュ・ラインは決して犠牲として選択されない。所与の合同クラスのマスク・ビットがすべて設定されてクラスの全てのメンバに欠陥があることが示された場合、システムは2ビットECC誤りの場合のように停止またはその他の操作で応答する。

【0025】修復マスク76は、欠陥キャッシュ・ラインが決してキャッシュ・ヒットを示さないようにする目的と、欠陥キャッシュ・ラインが決して犠牲として選択されないようにする目的の両方にとって好都合な手段であることがわかるであろう。したがって修復マスク76は、従来技術のキャッシュ構成要素で設けられるビット・ライン冗長構成及びワード・ライン冗長構成の代わりに使用することができる。修復マスクを追加するためにキャッシュ・チップ上にいくらかの余分な空間が必要であるが、この空間は一般に、ディレクトリ配列、LRU配列、及びキャッシュ配列内の冗長ビット・ライン及び冗長ワード・ラインをなくすことによって節約される空間量と比較すれば無視可能な程度である。この利点はキャッシュ・ライン・サイズが増大すると共に大きくなる。すなわち、キャッシュ・ライン・サイズの拡大に伴って修復マスク配列サイズが増大しない。また、修復マスク76を使用することにより、一部の（冗長）キャッシュ・ラインが一度も使用されなくなることはなくすべての使用可能キャッシュ・ラインが使用され、キャッシュの全体的使用効率が向上する。さらに修復マスク76はこれらの利点を、再経路指定オーバーヘッドなしに、ディレクトリ配列、LRU配列、またはキャッシュ配列の「ヒューズ切り」を必要とせずに得られるようにする。さらにこれによって、キャッシュ操作が格段に高速化し、製造コストが大幅に低減される。

【0026】機能的マスクを使用してキャッシュ内の欠陥を迂回するこの新規な方法により、パフォーマンス低下と標準のキャッシュ欠陥修復方法によるシリコン面積の増大がなくなる。機能の観点から見ると、特定の合同

クラスは（８ウェイ・セット・アソシアティブではなく）６ウェイまたは７ウェイのセット・アソシアティブを有効に稼働させることができる。しかし、キャッシュ動作の統計的な性質のために、このアソシアティブ性の減少はユーザ・レベルでは一般には目に付かない。

【００２７】さらに、修復マスクを使用すると、誤りが検出されたときに修復マスクをリアルタイムで更新することによって動的なキャッシュ欠陥迂回が可能になる。最初にキャッシュ・ラインを製作時に検査し、欠陥が見つかった場合は修復マスク内の対応するフィールドの値を永続的に設定することによってその欠陥を処理することができる。その後、コンピュータをブート（電源投入）するたびに、ブート・プロセスの一部としてファームウェア検査に基づいてマスクを自動的に更新することができる。最後に、修復マスクはディレクトリ・パリティ誤り、キャッシュ項目配列ＥＣＣ誤り、またはＬＲＵ誤りの検出時に更新することができる。修復マスク配列内の値を設定するハードウェア・アルゴリズムを備えることもできる。たとえば、各キャッシュ・ラインの修復マスク内に１つの２ビット・フィールドを設けることができる。この２ビット・フィールドを最初にゼロに設定し、そのキャッシュ・ライン上で誤りが検出されるたびに増分することができる。これによって、２ビット・フィールドはカウンタとして機能することができ、所与のキャッシュ・ラインについて３つの累積パリティ誤りが記録された場合にのみキャッシュ・ラインに欠陥があるものとして設定される。

【００２８】キャッシュ内のラインに関連づけられた修復マスク項目が設定されてそのラインに欠陥があることが示された場合、欠陥のあるキャッシュ場所に遭遇した後でプロセッサを信頼性をもって稼働させ続けるために、その場所にあるキャッシュの内容をフラッシュする。修復マスク項目が設定された後は、そのキャッシュ・ラインが将来アクセスされてもそのアクセスは修復マスクによってそのライン上で強制的にミスにされ、そのラインは二度と再使用（犠牲に）されない。この解決策は、冗長ラインなどの従来技術の方式と比較すると実質的にオーバーヘッドがない。これは、プロセッサが苛酷な環境で稼働しており、万一ランタイム欠陥が発生しても機能し続けなければならない応用分野で特に有用である。

【００２９】図３には１つのディレクトリ７２しか図示されていないが、本発明により構成されたキャッシュは図４に示すように追加のディレクトリ９６を有することもできる。ディレクトリ７２及び９６は冗長であるが、ディレクトリ７２はＣＰＵスヌープに使用され、ディレクトリ９６はシステム・バス・スヌープに使用される。言い換えると、各スヌーピング装置／相互接続機構に１つのディレクトリを備える。この構成によっていくつかの利点が得られる。第１に、両方のディレクトリを図３

に関連して説明したパリティ検査器を使用して構成した場合、各ディレクトリは他方のディレクトリのバックアップとして機能することができる。言い換えると、たとえばディレクトリ７２内のアドレス・タグ上でパリティ誤りが発生した場合、そのアドレス・タグを代わりにディレクトリ９６から読み取ることができる。

【００３０】パリティ誤りが発生した場合、図３に示すようなパリティ誤り制御（ＰＥＣ）装置９８を使用してその誤りを処理することができる。１００の接続線で示すように、ＰＥＣ装置９８は各パリティ検査器８４だけでなく、他方のディレクトリ９６のパリティ検査器にも接続されている。ＰＥＣ装置９８は最初にいずれかのパリティ検査器からパリティ誤りを検出すると、キャッシュを強制的にビジー・モードにし、その場合、誤りが処理されるまで要求が再試行されるか、または肯定応答されない。ＰＥＣ装置９８は次に、他方の（非エラー発生）ディレクトリ内の指定ブロックからアドレス・タグ（及び状態ビット）を読み取り、そのアドレス・タグを問題のあるディレクトリ、すなわち該当する比較器８２に直接供給する。問題のある配列を更新した後、ＰＥＣ装置８２はキャッシュが通常の操作を再開することができるようにする。

【００３１】本発明のＰＥＣ及びパリティ検査器の特に有利な一態様は、それらを使用して複数ビット誤り検出の一形態を提供することができることである。２４ビットのアドレス・タグを３個の８ビット・バイトに分けるなど、特定の値（アドレス・タグ）をいくつかの部分に分けることができる。次に、各部分についてパリティ・ビットを設ける。すなわち、この例では１つのアドレス・タグについて３つのパリティ・ビットを設ける。各部分がパリティ誤りを示すことになるため各部分の１ビットに誤りのある値がある場合でもその誤りは検出され、１つの部分のみのパリティ誤りだけでＰＥＣ９８に警告するのに十分である。その後、他方のディレクトリからそのビット・フィールド（アドレス・タグ）を代用することによって３つの誤りすべてが修正され、したがって本発明によって複数ビット誤りの修正が可能になる。

【００３２】図４で、「ＣＰＵスヌープ」と示されている線は一般にキャッシュのＣＰＵ側の相互接続からの操作を指し、ＣＰＵへの直接相互接続または他のスヌープ装置、すなわちより上位のキャッシュ（たとえばＬ１）への直接相互接続を含むことができる。「システム・バス・スヌープ」は一般にキャッシュのシステム・バス側の相互接続からの操作を指し、システム・バスへの直接相互接続または他のスヌープ装置、すなわち下位キャッシュ（たとえばＬ２）への直接相互接続を含むことができる。したがって本発明はキャッシュ階層の特定のレベルにもその全体的な深さにも限定されない。

【００３３】図４には、冗長キャッシュ・ディレクトリしか図示されていないが、幹線ＥＣＣ回路の代わりにパ

リティ検査器を備えた冗長キャッシュ項目配列（２個）も同様に使用することができ、その場合、１つのキャッシュ項目配列内に誤りがあると、他方のキャッシュ項目配列の対応するキャッシュ・ラインから値が読み取られることになる。この手法は、キャッシュのサイズを実際に２倍にする必要があるが、キャッシュ操作が高速化され、技術の進歩によってキャッシュ・サイズをますます小さくすることが可能になるに伴い、キャッシュの全体的サイズはその速度よりも重要ではなくなると考えられる。また、このようなキャッシュ・サイズの増加は、前述の修復マスクの使用によって生じるサイズの縮小によって部分的に相殺することができる。

【００３４】キャッシュにおいて２つのディレクトリを使用することによる他の利点は、１サイクルに２つの読取り操作を行うことができることである。すなわち、ＣＰＵスヌープからの１つの読取り操作とシステム・バス・スヌープからの１つの読取り操作を並列して行うことができる。従来技術のキャッシュ設計では所与のサイクルで１つの読取り操作しか行うことができないため、この特徴によってＣＰＵ及びシステム・バスからの全体的な読取りアクセス時間が大幅に短縮される。本発明では、両方の読取りを１クロック・サイクルで行うことができる。冗長ディレクトリの使用の唯一の考えられる欠点は、キャッシュ・ディレクトリ・サイズを２倍にする必要だけである。しかし、１サイクルで２つのスヌープ操作を行う能力に伴うパフォーマンスの向上があれば、このサイズの増大は許容可能なものである。さらに、クリティカル・パスからＥＣＣ回路を除去したことによって読取り操作がさらに高速で行われるため、キャッシュ速度をさらに向上させることができる。

【００３５】複数のスヌープ装置からの操作に独立して応答するように複数のキャッシュ・ディレクトリを設けることによる他の利点は、ＣＰＵスヌープかシステム・バス・スヌープかを選択するアービトレーション論理回路がなくなることである。アービトレーション論理回路は従来、クリティカル・パスにあるため、これによってアクセス時間が短縮されると共に、一般にキャッシュの複雑さが緩和される。２つのディレクトリを設ける他の利点は物理層に関する。単一ディレクトリを使用した場合、キャッシュ・チップ上のディレクトリから物理的に離れたところにあるいくつかのキャッシュ・ラインにはより長いアクセス時間を必要とする。たとえば、単一ディレクトリはそのようなアクセス時間の増加を最小限にするためにチップの中心付近に配置されることが多い。２つのディレクトリを設けることにより、それらをキャッシュ・チップ上で物理的に離して（すなわち中心部ではなく側面付近に）配置することができ、チップ上の導体経路を短縮することによって応答時間を短縮することができる。

【００３６】この二重読取り能力は、単一キャッシュ項

目配列でも２（冗長）キャッシュ項目配列でも備えることができる。後者の場合、２つのディレクトリ７２及び９６にそれぞれ接続された２つの異なる制御回路によって別々に制御される２つの異なるマルチプレクサを使用して、２つのキャッシュ項目配列からデータを読み取る。前者の場合も２つの異なるマルチプレクサを使用することができるが、それらは同じキャッシュ項目配列に接続することができる。すなわち、各キャッシュ・ブロックが２つのキャッシュ・ラインを有し、一方はＣＰＵスヌープのために第１のマルチプレクサに接続され、他方はシステム・バス・スヌープのために第２のマルチプレクサに接続される。

【００３７】図４のキャッシュにメモリ・ブロックを書き込むとき、アドレス・タグ（及び状態フィールドや包含フィールドなどの各種ビット）を両方のディレクトリ７２及び９６に書き込まなければならない。書込みは、ディレクトリ７２及び９６に接続された１つまたは複数の書込み待ち行列９４を使用して行うことができる。２つのディレクトリへの書込みは並列して行うことができる。しかし、これにはＣＰＵポートもシステム・バス・ポートも読取りを実行していないことを必要とし、したがって、どのサイクル中にもディレクトリのいずれか一方によって実行される読取り操作がある延長期間がある場合があるためにそのような並列書込みがキャッシュ・ディレクトリの書込み待ち行列から排出されるのにより長い時間がかかる可能性がある。したがって、２つのディレクトリへの書込みをずらすこともでき、これは複数（冗長）キャッシュ・ディレクトリを設けるに伴うもう一つの利点である。

【００３８】後者の実施態様では、たとえばＣＰＵがディレクトリ７２を介して読取り操作を行っている場合、同じサイクル中にシステム・バスがディレクトリ９６に対して書き込み操作を行うことができる。ディレクトリ７２への対応する書込み操作は、ＣＰＵスヌープが読取り（またはその他の非書込み）操作を行っていない次のサイクルまたはそれ以降のサイクルまで延期する

（書込み待ち行列に入れておく）ことができる。冗長ディレクトリを使用する場合、書込み操作をこのように分割することによってキャッシュ操作を高速化することができる。２つのキャッシュ・ディレクトリと共に別個の（冗長）キャッシュ項目配列を使用した場合、それらのキャッシュ項目配列にメモリ・ブロックを書き込む操作も同様にずらすことができる。

【００３９】このずらした書込みはディレクトリから別個の読取りを行う能力も補完する。たとえば、第１のサイクル中にディレクトリ７２に対して読取り操作が行われており、ディレクトリ９６への第１の書込みによるずらされた書込み操作が開始されたばかりであるシーケンスを考えてみる。第２のサイクル中に、ずらされた書込み操作はディレクトリ７２への書込みによって完了

し、ディレクトリ 96 に対してまったく無関係の読取りが行われる。したがって、2 サイクルで 2 つの読取り操作と 1 つの書込み操作が行われたことになる。この効果は、（たとえば第 1 のサイクル中に 2 つの読取り操作を行わせ、次に第 2 のサイクル中に両方のディレクトリへの並列書込みを行うことによって）ずらされた書込みを行わなくても偶然得られることもあるが、この機能を使用することによってスヌープ操作の実行の柔軟性が高くなり、それによってパフォーマンスがさらに向上する。

【0040】まとめとして、本発明の構成に関して以下の事項を開示する。

【0041】（1）コンピュータ・システムのプロセッサによって使用されるキャッシュにおいて欠陥を迂回する方法であって、各ビット・フィールドがキャッシュ内の複数のキャッシュ・ラインのうちのそれぞれ 1 つのキャッシュ・ラインに対応するビット・フィールドの配列を有する修復マスクを設けるステップと、キャッシュ内の特定のキャッシュ・ラインに欠陥があるものとして識別するステップと、修復マスク配列内の対応するビット・フィールドを設定して特定のキャッシュ・ラインに欠陥があることを示すステップと、修復マスク配列二の対応するビット・フィールドに基づいて欠陥のあるキャッシュ・ラインへのアクセスを防止するステップとを含む方法。

（2）防止する前記ステップが、欠陥のあるキャッシュ・ラインが決してキャッシュ・ヒットにならないように防止するステップを含む、上記（1）に記載の方法。

（3）欠陥のあるキャッシュ・ラインが決してキャッシュ置換のための犠牲として選択されないように防止するステップを含む、上記（1）に記載の方法。

（4）各合同クラスが複数のキャッシュ・ラインからなる複数の合同クラスのうちの 1 つに所与のメモリ・ブロックをマップするステップをさらに含み、欠陥のあるキャッシュ・ラインへのアクセスを防止する前記ステップの結果、欠陥のあるキャッシュ・ラインがそれぞれの合同クラスから実質的に除去される、上記（1）に記載の方法。

（5）欠陥のあるキャッシュ・ラインへのアクセスを防止する前記ステップによって、キャッシュが、冗長性をもたせるためにキャッシュ・ラインを確保しておくことなく欠陥のないすべてのキャッシュ・ラインを使用することができるようにする、上記（1）に記載の方法。

（6）キャッシュが、要求されたアドレスがキャッシュ・ラインの一部にあるアドレス・タグに対応するかどうかを判断する複数の比較器を有し、所与の比較器が、要求されたアドレスが前記アドレス・タグ部分と一致しない場合はイナクティブであるが、要求されたアドレスが前記アドレス・タグ部分と一致する場合はアクティブであり、欠陥のあるキャッシュ・ラインが決してキャッシュ・ヒットにならないように防止する前記ステップが、

各信号が比較器のうちの所与の 1 つの比較器の出力と修復マスクからのビット・フィールドのうちのそれぞれ 1 つのビット・フィールドの出力との AND 結合である複数の信号を入力信号として受け取るマルチプレクサを設けるステップを含む、上記（2）に記載の方法。

（7）キャッシュが、複数のキャッシュ・ラインのうちのどのキャッシュ・ラインを置換のための暫定的犠牲として選択するかを示す複数の出力信号を有するキャッシュ置換回路を有し、欠陥のあるキャッシュ・ラインが犠牲として選択されないように防止する前記ステップが、複数の信号を入力信号として受け取るマルチプレクサを設けるステップを含み、各信号が、暫定犠牲に欠陥がないように保証するために修復マスクのビット・フィールドから入力信号を受け取る代替犠牲選択論理回路によって変更されたキャッシュ置換回路の出力信号である、上記（3）に記載の方法。

（8）防止する前記ステップが、欠陥のあるキャッシュ・ラインが決してキャッシュ・ヒットにならないように防止するステップをさらに含む、上記（3）に記載の方法。

（9）プロセッサと、メモリ・デバイスと、前記プロセッサと前記メモリ・デバイスとに接続され、前記メモリ・デバイスのアドレスに対応するメモリ・ブロックを記憶する複数のキャッシュラインを有するキャッシュと、各ビット・フィールドが、選択されたキャッシュ・ラインに欠陥があることを示す値を有する前記選択されたキャッシュ・ラインに対応する前記ビット・フィールドのうちの所与の 1 つのビット・フィールドに基づいて前記複数のキャッシュ・ラインのうちの選択された 1 つのキャッシュ・ラインへのアクセスを防止する、前記キャッシュ・ラインのうちのそれぞれ 1 つのキャッシュ・ラインに対応する複数のビット・フィールドを有する修復マスクとを含むコンピュータ・システム。

（10）前記修復マスク手段が、欠陥のあるキャッシュ・ラインが決してキャッシュ・ヒットにならないように防止する、上記（9）に記載のコンピュータ・システム。

（11）前記修復マスク手段が、欠陥のあるキャッシュ・ラインが決してキャッシュ置換のための犠牲として選択されないように防止する、上記（9）に記載のコンピュータ・システム。

（12）前記キャッシュが、各合同クラスが複数の前記キャッシュ・ラインから成る複数の合同クラスの 1 つに所与のメモリ・ブロックをマップし、前記修復マスク手段が、欠陥のあるキャッシュ・ラインをその対応する合同クラスから有効に除去することにより、その欠陥のあるキャッシュ・ラインへのアクセスを防止する、上記（9）に記載のコンピュータ・システム。

（13）前記修復マスク手段によって、前記キャッシュがキャッシュ・ラインを冗長性をもたせるために確保し

ておくことなくすべての欠陥のないキャッシュ・ラインを使用することができるようにする、上記（９）に記載のコンピュータ・システム。

（１４）前記キャッシュが、要求されたアドレスがキャッシュ・ラインの一部にあるアドレス・タグに対応するかどうかを判断する複数の比較器を有し、前記比較器のうちの所与の１つの比較器が、前記要求されたアドレスが前記アドレス・タグ部分と一致しない場合はイナクティブであるが、前記要求されたアドレスが前記アドレス・タグ部分と一致する場合はアクティブである出力信号を有し、前記キャッシュが、複数の信号を受け取り、各信号が前記比較器の前記出力信号の１つと前記修復マスク手段から前記ビット・フィールドのうちのそれぞれ１つのビット・フィールドの出力信号とのAND結合である複数の信号を入力信号として受け取るマルチプレксаをさらに含む、上記（１０）に記載のコンピュータ・システム。

（１５）前記キャッシュが、複数の前記キャッシュ・ラインのうちのどのキャッシュ・ラインを置換のための暫定的犠牲として選択するかを示す複数の出力信号を有するキャッシュ置換回路を有し、各信号が前記修復マスク手段の前記ビット・フィールドから入力信号を受け取る代替犠牲選択論理回路によって変更された前記キャッシュ置換回路の出力信号を有する複数の信号を入力信号として受け取るマルチプレксаをさらに含む、上記（１１）に記載のコンピュータ・システム。

（１６）前記修復マスク手段が欠陥のあるキャッシュ・

ラインが決してキャッシュ・ヒットにならないようにさらに防止する、上記（１１）に記載のコンピュータ・システム。

#### 【図面の簡単な説明】

【図１】従来技術のマルチプロセッサ・コンピュータ・システムのブロック図である。

【図２】従来技術のセット・アソシアティブ・キャッシュを示す高水準略図である。

【図３】パリティ誤り制御回路と動的修復マスクとを有する、本発明により構成されたセット・アソシアティブ・キャッシュを示す高水準略図である。

【図４】２つの冗長ディレクトリを有する、本発明により構成されたキャッシュを示すブロック図である。

#### 【符号の説明】

７０ セット・アソシアティブ・キャッシュ

７２ キャッシュ・ディレクトリ

７４ LRU配列

７６ 修復マスク

７８ 制御論理回路

８０ 合同クラス

８２ 比較器

８４ パリティ検査器

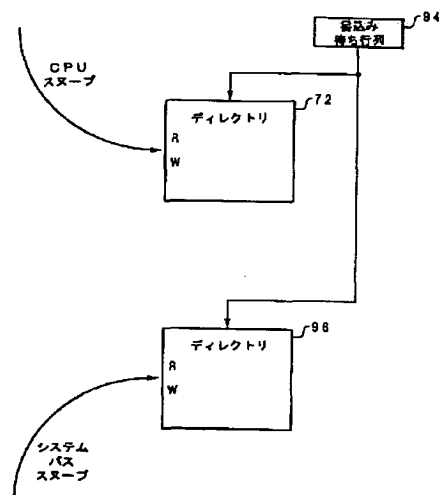
８６ ANDゲート

８８ マルチプレкса

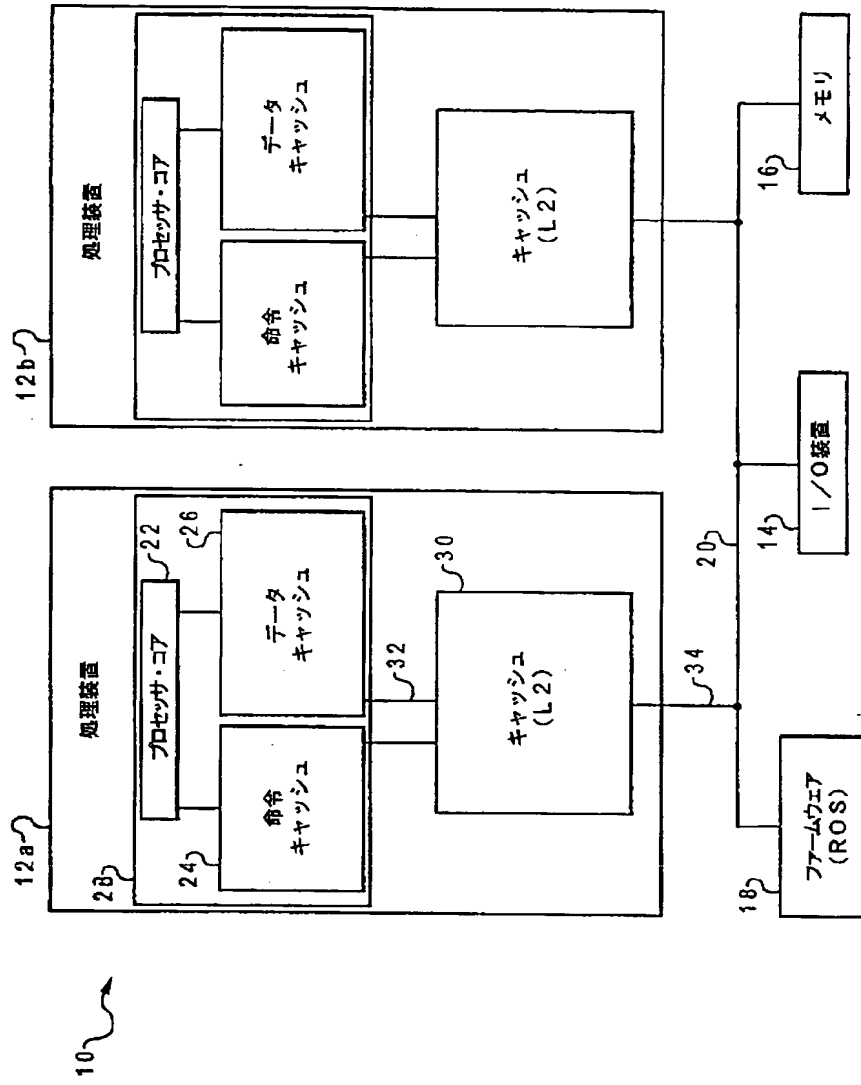
９０ ORゲート

９２ 代替犠牲選択回路

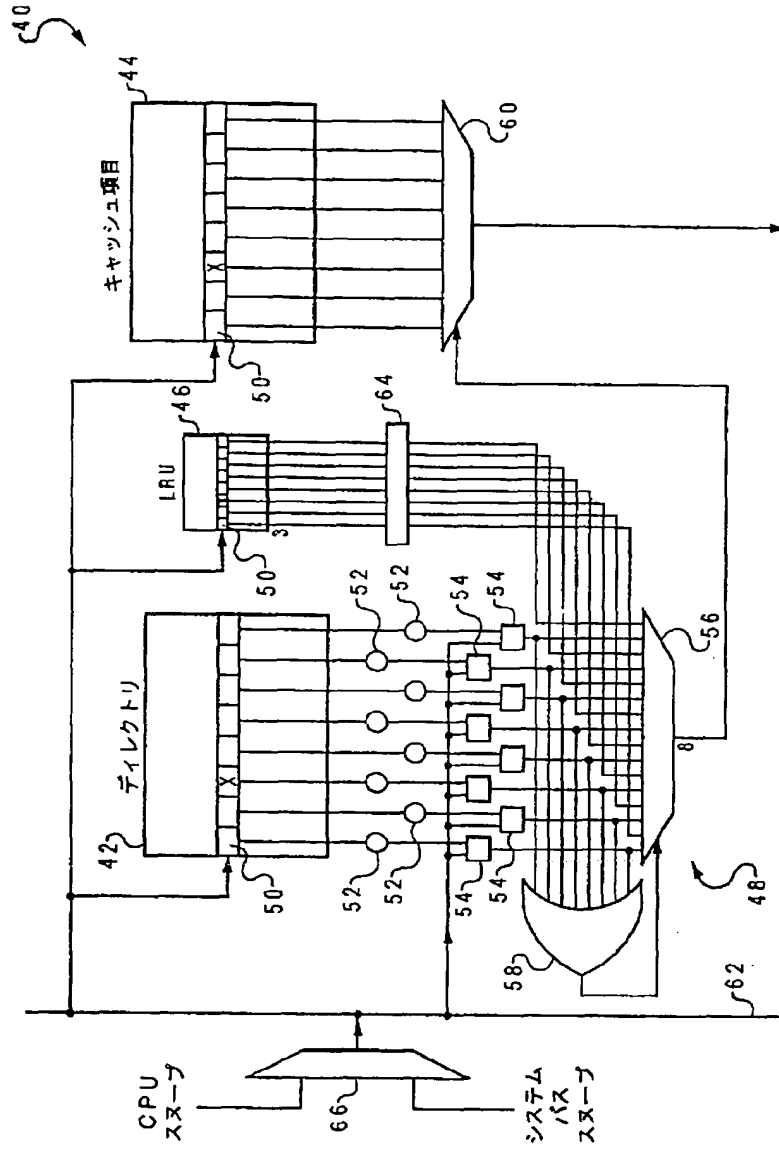
【図４】



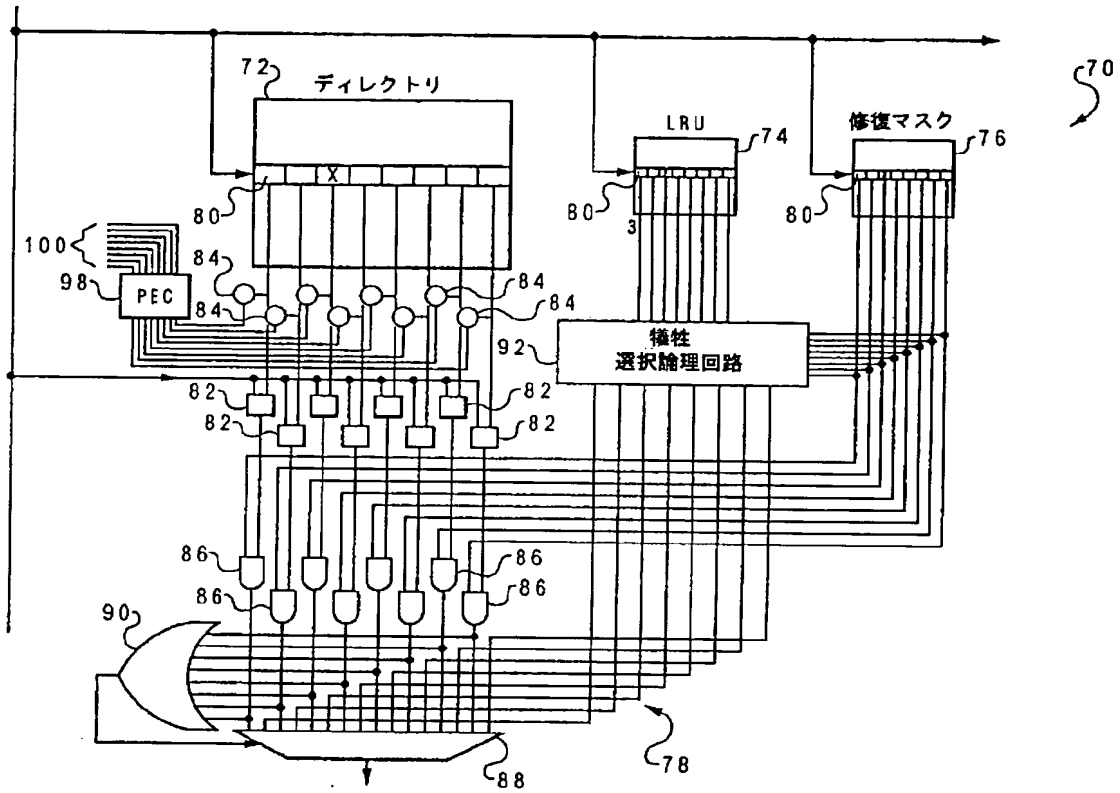
【図1】



【図2】



【図3】



フロントページの続き

(72)発明者 ジョン・スティーブン・ダッドソン  
アメリカ合衆国78660 テキサス州フラジ  
ヤービル ベル・ロック・サークル 1205

(72)発明者 ジェリー・ドン・ルイス  
アメリカ合衆国78681 テキサス州ラウン  
ド・ロック アローヘッド・サークル  
3409

(72)発明者 ティモシー・エム・スカーガン  
アメリカ合衆国78759 テキサス州オース  
チン バックソーン・ドライブ 10805